

Sistema Autónomo de Detección y Almacenamiento de Errores por Sucesos Aislados en SRAMs

Ángel Amigo, Carlos Palomar, Francisco J. Franco, Isabel López-Calle, María Fe Cervera, Antonio H. Cachero y Juan A. Agapito

Abstract—En esta contribución se explica el diseño, fabricación y verificación experimental de un sistema automático de lectura y escritura de memorias SRAM cuyo objeto es la determinación de la tasa de errores asociados a la acción de los rayos cósmicos e impurezas radiactivas en esa tecnología de fabricación. Se ha concebido este sistema para que se pudiera construir en una placa de reducidas dimensiones y que, de este modo, pudiera transportarse en la cabina de un avión durante vuelos comerciales.

Index Terms—Fiabilidad, circuitos integrados, sistemas autónomos

I. INTRODUCCIÓN

HOY en día vivimos rodeados de dispositivos electrónicos digitales. Muchos de ellos son de composición muy sencilla, con circuitos combinacionales, aunque la complejidad de los sistemas crece al resultar cada vez más fácil el acceso a dispositivos altamente integrados. Habitualmente, cuando empieza a aumentar la complejidad de los sistemas informáticos, surge la necesidad de almacenar datos, ya sea de forma temporal o persistente. Ejemplo de ello son las memorias de datos o, por ejemplo, los registros internos de un microprocesador o una FPGA. Es, por lo tanto, de gran importancia que el funcionamiento de los circuitos de memoria sea correcto durante el funcionamiento del dispositivo.

Los circuitos de memoria almacenan información de muy diversas formas, pero el objetivo final es siempre el mismo: poder recuperar dichos datos transcurrido cierto tiempo (a veces son apenas milisegundos, otras veces pueden pasar años). Es, por lo tanto, crucial la realización de estudios de integridad de datos en memorias, su fiabilidad, durabilidad, tasas de errores, etc. Uno de los grandes problemas a los que se enfrenta la electrónica actual consiste en la aparición de errores causados por elementos radiactivos: Estos pueden ser o partículas alfa resultantes de la desintegración de impurezas naturales o iones provenientes de la irrupción de un rayo cósmico [1]. Estos errores están asociados, principalmente, a elementos de memoria basados en dos inversores CMOS en anillo como son, por ejemplo, las celdas SRAM típicas. El paso de una partícula ionizante, sea cual sea su origen, induce un pulso de corriente en el elemento de memoria que, en determinadas circunstancias, provoca la sustitución de un dato por su opuesto. Este suceso se conoce como SEU, acrónimo

del término inglés *Single Event Upset*. A priori, no se sabe con claridad cual va a ser la evolución de la tasa de error a medida que se produce una mayor integración. Por un lado, se cree que la tasa de errores aumentará debido a que la carga eléctrica almacenada en cada celda es menor. Asimismo, es factible que varias celdas contiguas se vean afectadas por una única partícula creando lo que se conoce como error múltiple. Existen dos tipos de error múltiple: MBU (*Multiple bit upset*), si se producen varios errores en una única palabra lógica y MCU (*Multiple cell upset*) si ocurren en palabras distintas. Terminado este pequeño inciso, hay que indicar asimismo que el volumen de colección de carga, que es la zona donde se recogen portadores para el pulso de corriente, también disminuye y con él lo hace la posibilidad de que ocurra un evento. Por tanto, no se sabe con certeza la evolución esperable [2].

Esto ha llevado a tomar la siguiente determinación: Es necesario medir la tasa de error asociada a cada tecnología de fabricación. Uno de los métodos de evaluación típicos es la creación de bancos de memorias en las que se almacenan datos y se dejan funcionar durante meses o días para acumular datos estadísticamente significativos [3]. Para ello, es recomendable realizar los experimentos en laboratorios sitios a gran altitud pues el flujo de rayos cósmicos aumenta a medida que nos alejamos del nivel del mar. *Grosso modo*, se puede estimar que aumenta 10 veces cada 3000 m de altura. Esta estrategia ha sido llevada a cabo por Xilinx en el proyecto Rosetta [4], por un equipo de la Universidad de Aix-Marsella en el Pic de Bure, en los Alpes franceses, etc [5]. Otra estrategia, parecida, consiste en la elaboración de un banco de memorias suficientemente pequeño para transportarlo en vuelos comerciales [6], [7]. Esta estrategia se ha demostrado eficaz para tecnologías de 130 nm como muestra el trabajo del equipo ARIS del laboratorio TIMA-IMAG de Grenoble (Francia) y es el punto de partida de este trabajo: Construir una placa similar utilizando microprocesadores en lugar de FPGAs y examinar de manera automática memorias mientras se transportan en la cabina de un avión comercial. Durante la mayor parte del vuelo, las memorias se encuentran a más de 10 km de altura con un flujo de radiación natural excepcionalmente alto.

Finalmente, es recomendable repetir experimentos en laboratorios a ras de suelo o, mejor aún, a gran profundidad para poder discriminar si los fenómenos observados se deben a la acción de los rayos cósmicos o las impurezas radiactivas [8], [9].

Este trabajo fue financiado parcialmente por el proyecto MCINN AYA2009-13300-C03-03 y por UCM-BSCH.

Todos los autores pertenecen al Departamento de Física Aplicada III, Facultad de Física, Universidad Complutense de Madrid (UCM), 28040 Madrid (Spain) (email: fjfranco@fis.ucm.es).

II. OBJETIVOS

El principal objetivo de este trabajo era la creación de un banco de memorias portátil sobre el que pudiésemos hacer estudios y pruebas variadas que se realizasen de forma automática y recurrente. El principio de funcionamiento es sencillo. Al encender el banco de memorias, se almacena un patrón de datos en los módulos que componen el banco y, cada cierto tiempo, de forma automática, se comprueban los valores leídos y se comparan con los que se habían escrito. Se decidió que el patrón de escritura fuese seleccionable desde el ordenador personal (PC) de control. Así, podía escribirse cualquier patrón de 8 bits enviado desde el PC en todas las direcciones o alternando un patrón con su complementario, se podían escribir datos de un modo pseudoaleatorio, etc. Hay que subrayar que este paso no es estrictamente necesario pues se concibió el sistema para realizar la toma de datos independientemente de haber recibido instrucciones del ordenador. Así, por defecto, el sistema escribía el patrón 0x00 en las direcciones impares y 0xFF en las pares.

En principio, se decidió que el programa chequeara el banco de memorias cada media hora para, así, tener una distribución temporal de eventos. Si se detecta algún error (alguno de los datos leídos no se corresponde con el dato que se había escrito en la inicialización), se almacena toda la información relativa al error (módulo, dirección de memoria, dato esperado y dato leído) en una memoria persistente, que posteriormente se podrá leer desde un ordenador una vez finalizado el experimento. Se decidió que esta memoria perteneciera a la familia de las memorias RAM magnetorresistivas (MRAMs), en las que los datos se guardan sin recurrir al almacenamiento de carga eléctrica por lo que, a priori, son insensibles a errores asociados a la radiactividad natural.

Se concibió el diseño de tal modo que pudiera ser alimentado con baterías. En principio, el sistema de baterías permitía alimentar los dispositivos a su tensión nominal de trabajo (3.3 V). Sin embargo, se añadieron entradas de alimentación adicionales para poder alimentar el sistema con tensiones de trabajo menores y así aumentar las aplicaciones del banco de memorias. La tensión de alimentación es un parámetro crítico, como se ha descubierto en dispositivos parecidos, y es conveniente introducir este grado de libertad [2].

III. DESCRIPCIÓN DEL SISTEMA

A. Bloque de alimentación

Se optó por alimentar el sistema con un regulador lineal de tensión TPS76633, de Texas Instruments. Este regulador proporciona una tensión estable de alimentación de 3.3 V, que es la tensión nominal de funcionamiento de las memorias. El módulo de alimentación puede ir conectado tanto a una batería como a una fuente de alimentación externa alternativa. Utilizando una batería de NiMH con una tensión de salida de 7.2 V y una carga de 4000 mA·h, se conseguía una autonomía de más de una semana de funcionamiento ininterrumpido a temperatura ambiente. Se comprobó experimentalmente que el consumo de corriente era del orden de 15-20 mA (Dependiendo de la tensión de alimentación) durante las fases

de escritura y lectura, que no duraban más de 10 minutos. Durante el resto del tiempo, el consumo era despreciable.

Asimismo, se incluyó una entrada adicional para añadir una tensión de alimentación distinta y así, de este modo, investigar la tasa de errores a distintas tensiones de alimentación. Se sabe que cuanto menor sea la tensión de alimentación mayor es la posibilidad de que se produzca un suceso aislado. Una limitación importante del diseño son los distintos valores de tensión mínima de alimentación de cada uno de los bloques. En la práctica, el sistema está limitado por la tensión mínima de alimentación de la memoria MRAM de salvaguarda. Esta memoria, del tipo MR256A08B, tiene nominalmente una tensión mínima de funcionamiento de 3.0 V aunque, en realidad, pudimos constatar que el valor real de esta tensión era 2.8 V. Los otros elementos clave de la placa (microprocesador y memorias) tienen tensiones mínimas de funcionamiento del orden de 2.0 V. Esto implica que, si bien puede intentarse el estudio con tensiones menores, hay que modificar el software de la placa para evitar el uso de esta memoria de salvaguarda.

B. Bloque principal

El almacenamiento principal a probar está compuesto por 32 módulos de memoria de la marca Cypress Semiconductors, modelo CY62167EV30. Son módulos de 16 Mbits, por lo que la capacidad total de memoria de la placa es de 512 Mbits. Dicha capacidad se podía estructurar como 1M de palabras de 16 bits o como 2M de palabras de 8 bits. En nuestro caso, optamos por la configuración de palabras de 8 bits. El motivo de esta elección era, simplemente, simplificar la construcción de la placa a expensas de doblar la duración del ciclo de lectura y escritura.

En cualquier caso, la anchura del bus de direcciones estaba entre 16 y 24 lo que imponía una restricción adicional en el microprocesador de control. Otro punto especialmente interesante es que, en el momento de construcción del sistema, Cypress Semiconductors es el único fabricante que ofrecía memorias SRAM con buses de datos y direcciones en paralelo y construidos en tecnología CMOS de 90 nm. Por el momento, ningún otro fabricante ofrece memorias a una escala menor a pesar de que otros dispositivos CMOS como FPGAs se ofrecen en tecnología de 22 nm. Otra ventaja adicional consistía en que estas memorias se proporcionan con un encapsulado TSOP-I, muy manejable en un laboratorio en comparación con otros formatos como el tipo VFBGA.

La lectura y escritura del banco de memorias se realizaba con el PIC18F85J90, fabricado por Microchip. Se seleccionó este dispositivo por la experiencia del grupo en microprocesadores de este tipo. Asimismo, es compatible con las tensiones de alimentación típicas de las memorias y, por otra parte, ofrecía una gran cantidad de puertos de entrada y salida, necesarios para este sistema. Este microprocesador era controlado por un cristal de cuarzo de frecuencia de resonancia igual a 20 MHz. La selección de la memoria que era escrita o leída en cada instante se llevaba a cabo con dos decodificadores SN74HC154DW, de Texas Instruments. La elección de decodificadores se realizó por dos claros motivos: En primer lugar, se evitaba tener que reservar 32 pines del

microprocesador para habilitar y deshabilitar memorias. Así, solo se necesitaron 6. Por otro lado, se facilitaba el rutado de la placa pues es más sencillo dividir las memorias en subconjuntos y llevar el decodificador y sus cuatro bits de control cerca de éstas, evitando tener que llevar un bus de 32 pistas.

El almacenamiento de la información acerca de los errores detectados era un punto crítico por varios motivos. En primer lugar, sería un hecho terrible que, tras realizar un experimento, los datos se perdieran al cesar la alimentación por una manipulación descuidada o por agotamiento de las baterías. Por otra parte, sería insensato almacenar datos en una memoria que fuera sensible a errores. Por ello, se decidió utilizar la memoria RAM magnetorresistiva MR256A08B, de Everspin, con una capacidad de 32K x 8 bits (un total de 262,144 bits). Dicha capacidad de almacenamiento era más que suficiente, ya que por cada error se reservaban 6 bytes, por lo que podíamos almacenar más de 5400 errores sin reinicializar la memoria (un valor muy superior a lo que esperábamos detectar). Cada error se registraba por medio de 6 bytes. El byte 1 almacenaba la memoria donde se había observado el evento. Los tres siguientes (2, 3 y 4) almacenaban la dirección, el 5 se reservaba para guardar el patrón erróneo leído y el 6 para almacenar el número de ciclo de lectura en el que se había observado. Por seguridad, y dada la gran capacidad de la memoria, se almacenaba la información por triplicado.

Esta memoria es inmune a fallos, como se ha demostrado en la literatura relacionada. Asimismo, se reservaron las primeras direcciones de la memoria para indicar datos generales como el número de ciclos de lectura, el patrón almacenado en el experimento y el número de eventos detectados. Finalmente, debe señalarse que esta memoria compartía el bus de direcciones y de datos con las memorias investigadas. Solamente la entrada de habilitación era específica de esta memoria.

C. Comunicación con el usuario

La comunicación con el ordenador se requería para realizar alguna modificación de la configuración y para comprobar la existencia de errores. Se llevaba a cabo por medio del protocolo RS-232, para lo que fue necesario integrar en la placa un dispositivo de la familia MAX3221, de Maxim Integrated. La razón para recurrir a este formato y no a otros más modernos, como el protocolo USB, radicaba en la extrema facilidad de programación. Asimismo, cabe recordar que no se esperaba un flujo de datos abrumador de modo que era perfectamente factible utilizar este protocolo.

Para conseguir la recuperación de datos, bastaba la utilización del programa Hiperterminal, disponible por defecto en la mayor parte de las distribuciones del sistema operativo Microsoft Windows. No fue necesario crear un programa específico con lenguajes de programación orientados a la instrumentación.

Por otra parte, se incorporó al diseño un juego de diodos emisores de luz (LEDs) con propósitos diversos. En primer lugar, indicar a cualquier usuario que la placa se encontraba en funcionamiento y que no se encontraba parada por el motivo que fuera. Asimismo, indicaba en qué proceso se encontraba

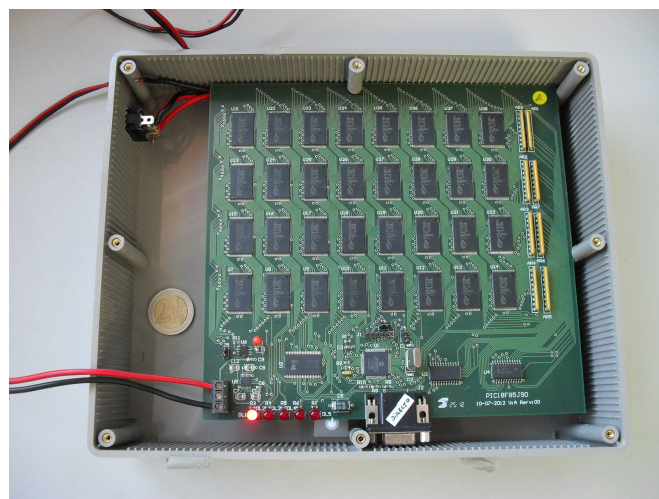


Figure 1. Fotografía del banco de memoria. Se ha colocado una moneda de dos euros para dar una clara idea del tamaño. El LED encendido muestra que se encuentra en proceso de escritura del patrón tras el encendido inicial. El espacio vacío de la parte izquierda se reserva para colocar la batería y, a continuación, colocar y atornillar la tapa.

la placa: Bien la escritura inicial, bien los ciclos de lectura, bien situación de espera. Para terminar, y como puede parecer lógico, se incorporó un LED que se activaba en caso de encontrarse al menos un error.

Fig. 1 muestra una fotografía del banco de memorias. En esa imagen, puede apreciarse con claridad la disposición de los distintos buses de datos, dirección y habilitación. A pesar de la gran complejidad del diseño y de los duros requerimientos topológicos, la placa pudo ser diseñada y fabricada utilizando únicamente cuatro capas, con el consiguiente abaratamiento en la fabricación.

IV. FLUJO DEL PROGRAMA

Tal y como se había explicado anteriormente, el funcionamiento de la placa es muy sencillo. El diagrama de flujo del algoritmo principal es el mostrado en Fig. 2. La descripción del funcionamiento es la siguiente:

- 1) Al encender el banco de memorias, se inicializan todos los puertos, para definirlos como entrada, salida o entrada/salida
- 2) Se inicializan los 32 módulos de memoria con el patrón seleccionado. Por defecto, el banco está programado para escribir 0x00 en los bytes pares y 0xFF en los impares. Dichos patrones se pueden modificar por medio de un cable serie, conectado al banco de memorias
- 3) Se leen varios datos de interés de la memoria MRAM (en los primeros bytes de la memoria se almacena un puntero indicando dónde se almacenarán los datos del siguiente error, para evitar sobrescribir datos al reiniciar el bando) y se inicializan las variables internas de funcionamiento del banco
- 4) Se activa un temporizador y se entra en un bucle infinito
- 5) Cuando el temporizador cumple el tiempo para el que estaba armado, se realiza una comprobación de la información almacenada en las memorias

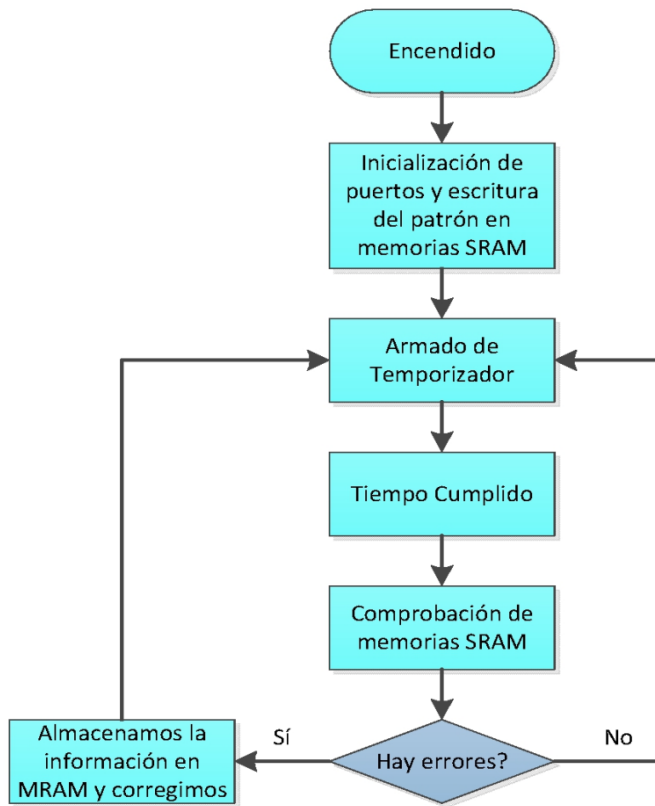


Figure 2. Flujo del programa para inicialización y detección de errores.

- 6) Como se conoce el patrón escrito inicialmente, si los datos leídos no coinciden con los esperados, se almacena la información de interés en la memoria MRAM (en nuestro caso, almacenábamos el número de comprobación que estábamos haciendo, el módulo en el que se produce el error, los 3 bytes de dirección de memoria del error, el patrón escrito y el valor erróneo leído). Posteriormente, se corrige el error, almacenando el patrón correcto en dicha dirección de memoria
- 7) Una vez comprobados todos los módulos de memoria, almacenado y corregido todos los errores detectados, se volvía a armar el temporizador, de forma que volveríamos a encontrarnos en el punto 4.

Otras funcionalidades que se incorporaron al programa y que tenían como objeto la depuración del código y verificación del sistema fueron las siguientes. En primer lugar, se añadió un sistema de LOG, de forma que si se activaba, la placa informaba por medio de la conexión RS232 de cada una de las funciones que estaba realizando. Esto resultó muy útil principalmente para las pruebas en laboratorio, y también nos pareció de mucha utilidad para poder comprobar el estado de las memorias y funcionamiento del banco desde una ubicación remota, ya que sólo resulta necesario tener un portátil y un cable serie. Por otra parte, se añadieron rutinas para inyección de errores en posiciones conocidas del banco de memoria con el objeto de verificar el sistema de detección, almacenamiento y envío. Asimismo, se incluyeron rutinas que permitían el volcado completo de la memoria MRAM y su borrado.

V. RESULTADOS

El banco de memorias se transportó en vuelos comerciales, acumulando un total de 27 vuelos. Durante estos vuelos, el banco estuvo encendido durante 90 horas, de las cuales 55 fueron horas de vuelo. Estos vuelos cubrían rutas entre distintos puntos de Europa con una duración típica de unas dos horas. Por término medio, se estimó que el flujo equivalente en cada vuelo estaba entre 155 y 175, dependiendo de la actividad solar. Estas cifras indican que la radiación ambiental era tantas veces la radiación cósmica recibida en la ciudad de Nueva York, a unos 45° de latitud y al nivel del mar. En otras palabras, estas 55 horas de vuelo equivalían a 9075 ± 550 horas a ras de tierra (Más de un año).

En estos vuelos, se pudo confirmar la detección de un error en un vuelo desde Amsterdam a Madrid a 39.000 pies de altitud (aproximadamente 12 km). Fue un error de bit en una dirección impar. Se esperaba haber leído el valor 0x00 (0b00000000) y se obtuvo el valor 0x04 (0b00000100). Ciertamente, es muy difícil obtener información relevante de un único evento aunque nos permite acotar la tasa de error esperable en este tipo de memorias. La aparición de eventos es un fenómeno describable mediante una distribución de Poisson, en la que la probabilidad de que un evento ocurra n veces en un intervalo de tiempo es

$$P(n) = e^{-\mu} \cdot \frac{\mu^n}{n!},$$

donde μ es la media de eventos esperados. El hecho de que el evento haya ocurrido una única vez implica que la probabilidad de que n sea igual a 1 no es despreciable. Por convenio, supondremos que $P(1) = \mu \cdot e^{-\mu} \geq 0.05$ con lo que $0.053 \leq \mu \leq 4.45$. En este tipo de problemas, es común utilizar una unidad llamada FIT (*Failures In Time*), que es número medio de eventos esperables en la ciudad de Nueva York al cabo de 10^9 h y observar 1 Mbit de datos. Aplicando factores de conversión sencillos, se puede concluir que la tasa de errores está entre 11 y 1020 FIT. A pesar de la amplitud del rango de valores, la tasa de error está en el orden de magnitud esperado en tecnologías de este orden.

Posteriormente, se examinaron las memorias durante dos meses en el laboratorio del grupo de investigación. En este caso, no se observó ningún evento lo que nos permitió deducir que el fenómeno observado fue probablemente debido a un rayo cósmico y no a una impureza radiactiva. El motivo es que, en este último escenario, la tasa de errores no depende de la altitud. En el caso más optimista, tendría que haberse observado de media 1.6 eventos y la distribución de Poisson solo ofrece una probabilidad del 20.2%. Esta probabilidad parece inclinar la balanza a favor de la acción de un rayo cósmico pero es necesario acumular más datos para poder afirmarlo con seguridad.

Asimismo, debe indicarse que las pruebas continuaron en el laboratorio sito en la Universidad Complutense de Madrid polarizando las muestras con una tensión de alimentación de 2.8 V. Como se dijo anteriormente, ésta es la tensión mínima de funcionamiento. En el momento de escribir este texto, no se volvió a registrar efectos. Dada la situación de Madrid, a cerca de 700 m de altura, se ha estimado que la tasa de error

de este tipo de memorias no puede ser mayor de 2440 FIT con una confianza del 95%. Este resultado es compatible con los datos registrados en vuelo.

Curiosamente, este resultado discrepa de los datos recogidos por el grupo ARIS de Grenoble, que determinaron que la tasa de error de estas memorias era del error de 7000 FIT [10]. Sin embargo, este hecho es fácilmente comprensible pues las memorias utilizadas en aquel experimento se compraron en 2007 en tanto que las nuestras fueron adquiridas mucho después, en 2011. En ese intervalo de tiempo, se habían publicado ya algunos resultados preliminares que apuntaban a la altísima sensibilidad de estas memorias lo que, probablemente, condujo a un rediseño interno de los componentes con el objeto de minimizar la tasa de errores esperable [11].

VI. AGRADECIMIENTOS

Angel Amigo desea agradecer a Air Europa, donde actualmente trabaja como piloto civil, todas las facilidades ofrecidas para trasladar la placa de pruebas en el interior de la cabina de las aeronaves.

VII. CONCLUSIÓN Y FUTURO TRABAJO

En este artículo, se ha demostrado la viabilidad de fabricación de un banco portable de memorias SRAM para deducir la tasa de error asociada a partículas radiactivas y se ha demostrado experimentalmente su correcto funcionamiento. En un futuro, se seguirá acumulando datos para poder elaborar una estadística fiable a partir de los datos obtenidos de este sistema. Asimismo, se pretende construir sistemas análogos a medida que vayan apareciendo en el mercado nuevos modelos de memorias SRAM en tecnologías más avanzadas, como 65 nm, 45 nm, etc.

REFERENCES

- [1] R. Velazco and F. Franco, "Single Event Effects on Digital Integrated Circuits: Origins and Mitigation Techniques," in *IEEE International Symposium on Industrial Electronics*, 2007. ISIE 2007., 2007, pp. 3322–3327.
- [2] M. Nicolaidis *et al.*, *Soft Errors in Modern Electronic Systems*, ser. Frontiers in Electronic Testing. Springer, 2011.
- [3] J. Ziegler, *SER-History, Trends, and Challenges: A Guide for Designing with Memory ICs*. Cypress Semiconductors, 2004.
- [4] A. Lesea, S. Drimer, J. Fabula, C. Carmichael, and P. Alfke, "The rosetta experiment: atmospheric soft error rate testing in differing technology FPGAs," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 317–328, 2005.
- [5] J. Autran, P. Roche, J. Borel, C. Sudre, K. Castellani-Coulie, D. Munteanu, T. Parrassin, G. Gasiot, and J.-P. Schoellkopf, "Altitude SEE Test European Platform (ASTEP) and First Results in CMOS 130 nm SRAM," *IEEE Transactions on Nuclear Science*, vol. 54, no. 4, pp. 1002–1009, 2007.
- [6] P. Peronnard, R. Velazco, and G. Hubert, "Real-Life SEU Experiments on 90 nm SRAMs in Atmospheric Environment: Measures Versus Predictions Done by Means of MuscaSep ? Platform," *IEEE Transactions on Nuclear Science*, vol. 56, no. 6, pp. 3450–3455, 2009.
- [7] L. Artola, R. Velazco, G. Hubert, S. Duzellier, T. Nuns, B. Guerard, P. Peronnard, W. Mansour, F. Pancher, and F. Bezerra, "In Flight SEU/MCU Sensitivity of Commercial Nanometric SRAMs: Operational Estimations," *IEEE Transactions on Nuclear Science*, vol. 58, no. 6, pp. 2644–2651, 2011.
- [8] S. Martinie, J. Autran, D. Munteanu, F. Wrobel, M. Gedion, and F. Saigne, "Analytical Modeling of Alpha-Particle Emission Rate at Wafer-Level," *IEEE Transactions on Nuclear Science*, vol. 58, no. 6, pp. 2798–2803, 2011.
- [9] S. Martinie, J. Autran, S. Sauze, D. Munteanu, S. Uznanski, P. Roche, and G. Gasiot, "Underground Experiment and Modeling of Alpha Emitters Induced Soft-Error Rate in CMOS 65 nm SRAM," *IEEE Transactions on Nuclear Science*, vol. 59, no. 4, pp. 1048–1053, 2012.
- [10] G. Hubert, R. Velazco, C. Federico, A. Cheminet, C. Silva-Cardenas, L. V. E. Caldas, F. Pancher, V. Lacoste, F. Palumbo, W. Mansour, L. Artola, F. Pineda, and S. Duzellier, "Continuous High-Altitude Measurements of Cosmic Ray Neutrons and SEU/MCU at Various Locations: Correlation and Analyses Based-On MUSCA SEP3," *IEEE Transactions on Nuclear Science (In Press)*, 2013.
- [11] R. Velazco, "Private Communication."